

목차

table of contents

1 주제 선정

2 분석 방법 설정

3 CRNN 모델이란?

4 1차 시도: 범용적 모델

5 2차 시도: 특화 모델

6 실제 적용: 멜론티켓 매크로 제작

7 부록: 보안 측면에서의 고찰

8 참고 문헌

1

주제 선정

CAPTCHA vs 자동화 매크로

가장 강한 방패 vs 가장 강한 창의 대결

Please complete the challenge below

I'm not a robot


reCAPTCHA
[Privacy](#) - [Terms](#)



주제 선정

What is CAPTCHA?

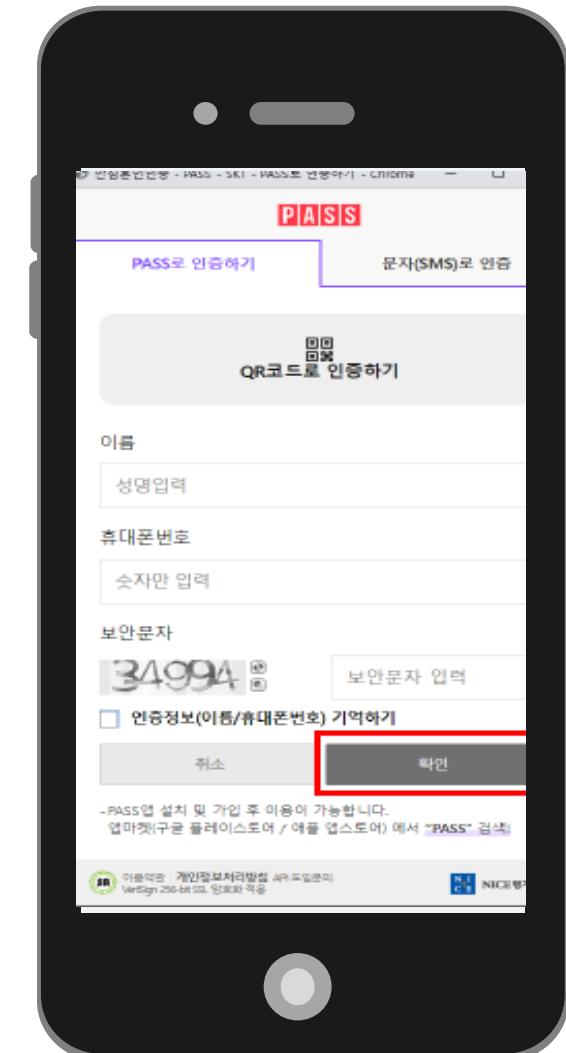
Completely Automated Public Turing test to tell
Computers and Humans Apart

이미지 혹은 오디오를 주고 정답을 맞히게 함으로써
사람과 컴퓨터를 판별

매크로를 개발 및 사용하고자 하는 입장에서는
파훼해야 하는 대상,
악성적인 공격과 매크로 이용을 막아야 하는
입장에서는 절대 **파훼되어서는 안 되는 보안장치**

튜링 테스트

질의자가 인간과 컴퓨터를 대상으로 정해진 시간 안에 대화를
나누는 방식으로 이루어지는데, 대화를 통해 인간과 컴퓨터를
구별해내지 못하거나 컴퓨터를 인간으로 간주하게 된다면
해당 기계는 인간처럼 사고할 수 있는 것으로 보는 인공지능 판별법



2

분석 방법 설정

분석 방법 설정

1. 각종 데이터를 이용하여
CAPTCHA를 해석할 수 있는 모델 훈련

2. 실제 CAPTCHA 이미지로 예측 & 성능평가

3. 다양한 개선 방안과 방법론을 적용해 보며
정확도 개선 시도

OCR model for reading Captchas

Author: [A_K_Nain](#)

Date created: 2020/06/14

Last modified: 2020/06/26

Description: How to implement an OCR model using CNNs, RNNs and CTC loss.

ⓘ This example uses Keras 3

[View in Colab](#) · [GitHub source](#)

Introduction

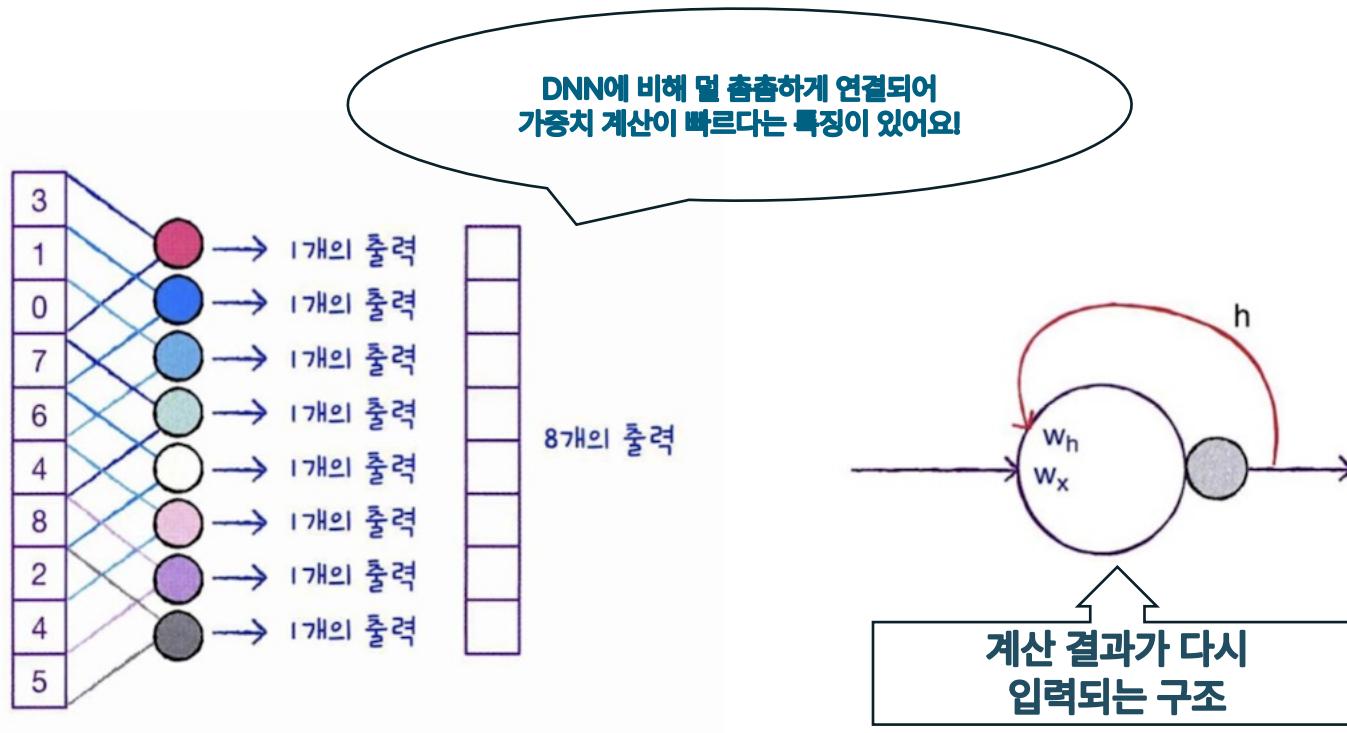
This example demonstrates a simple OCR model built with the Functional API. Apart from combining CNN and RNN, it also illustrates how you can instantiate a new layer and use it as an "Endpoint layer" for implementing CTC loss. For a detailed guide to layer subclassing, please check out [this page](#) in the developer guides.

Keras에서 제공하는 OCR 분석 방식인 CRNN 모델을 활용하기로 결정

3

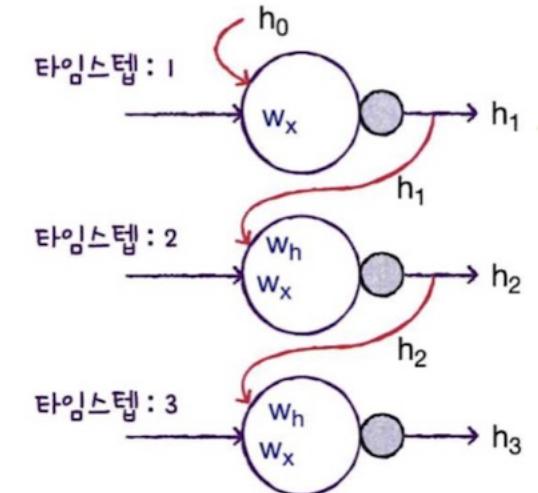
CRNN 모델이란?

CRNN 모델이란?



Convolution Layer를 사용하여
이미지의 특징을 학습하는 CNN

순환적인 구조를 활용하여
순차적인 패턴을 학습하는 RNN



→ 두 방식의 특징을 활용하여 이미지 내의 텍스트 인식에 활용할 수 있음

4

1차 시도: 범용적 모델

범용적인 Captcha 만들어보기



PARSA SAMADNEJAD · UPDATED 3 YEARS AGO

CAPTCHA Dataset

+113,000 alphanumeric colorful images

Data Card

Code (11)

Discussion (0)

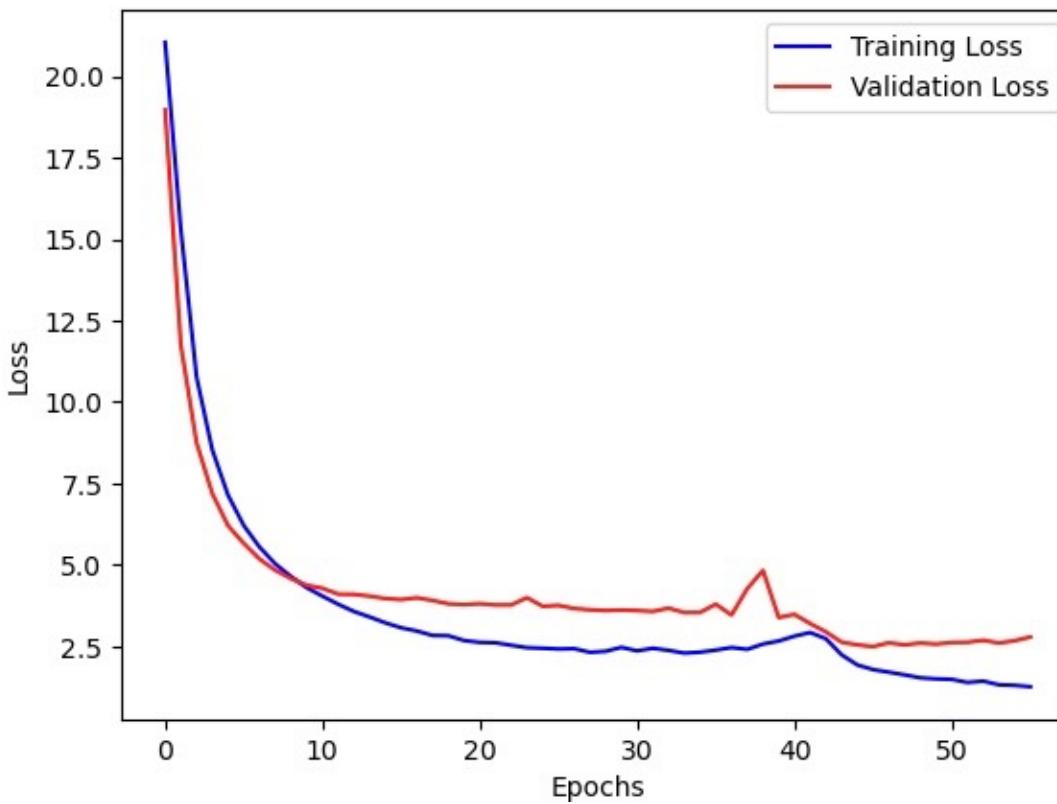
About Dataset

Content

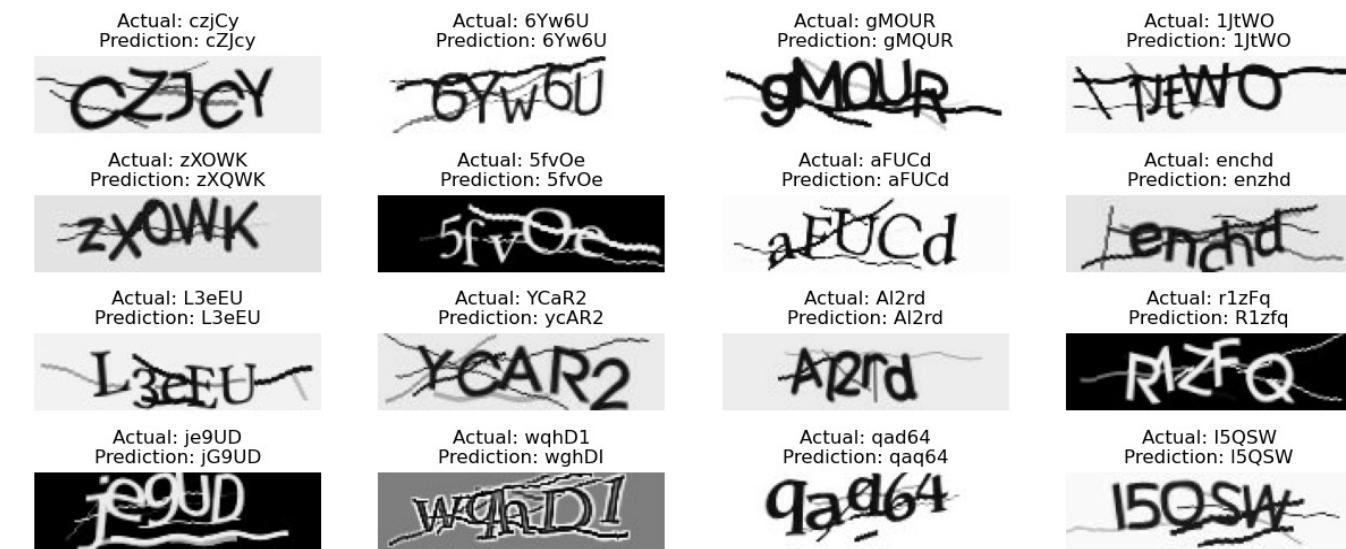
Captcha Dataset contains more than 113,000 colorful 5-character images.

- Kaggle에 있는 약 113,000개의 5글자 CAPTCHA를 Dataset으로 사용
- Keras의 코드와 다르게 Batch에 대한 정규화를 진행
→ 과적합 방지, 학습 속도 개선 목표

범용적인 Captcha 만들어보기



- Batch size = 32 (with normalization)
- Learning Rate = 0.001
- Validation loss = 1.9282



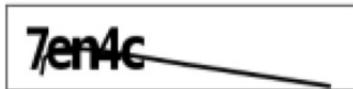
- 대소문자의 형태 차이가 크기에 불과한 경우, 예측에 실패하는 경우 존재(YCAR2, R1zfq)
- 애초에 레이블링이 잘못된 경우도 존재함(R1zfq, czjcy)

**사이즈도 크고... 꽤나 다양한 데이터가 모인 것 같은데
다른 사이트의 CAPTCHA에서도 뚫릴 수 있을까?**

범용적인 Captcha 만들기

- 위메프

Prediction : [UNK][UNK][UNK][UNK][UNK]
Actual : 7en4c



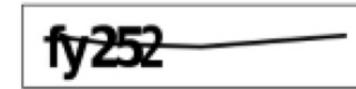
Prediction : c[UNK]4[UNK]b
Actual : 8b3g7



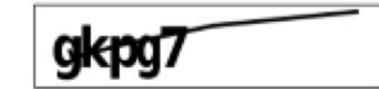
Prediction : [UNK]3[UNK][UNK][UNK]
Actual : b2chc



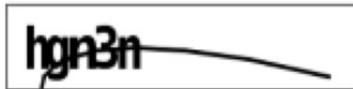
Prediction : 33[UNK][UNK][UNK]
Actual : fy252



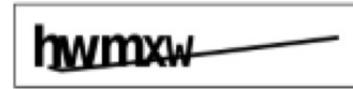
Prediction : [UNK]b[UNK][UNK][UNK]
Actual : gkpg7



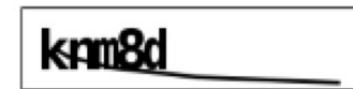
Prediction : [UNK][UNK][UNK][UNK][UNK]
Actual : hgn3n



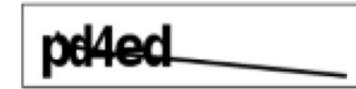
Prediction : [UNK][UNK][UNK][UNK][UNK]
Actual : hwmxw



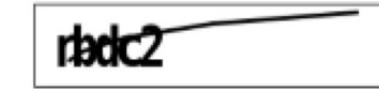
Prediction : [UNK][UNK][UNK][UNK][UNK]
Actual : knm8d



Prediction : [UNK][UNK][UNK][UNK][UNK]
Actual : pd4ed



Prediction : [UNK][UNK][UNK][UNK][UNK]
Actual : rbdc2



- 스타벅스

Prediction : n[UNK][UNK]2[UNK]
Actual : 5crhh



Prediction : a4[UNK]b[UNK]
Actual : 73n8g



Prediction : b[UNK][UNK][UNK][UNK]
Actual : 8gyef



Prediction : [UNK][UNK][UNK][UNK][UNK]
Actual : affnr



Prediction : [UNK][UNK]55[UNK]
Actual : pb4x8



Prediction : [UNK][UNK][UNK]a[UNK]
Actual : wnhc7



Prediction : [UNK][UNK][UNK][UNK]a
Actual : xhgm7



Prediction : [UNK][UNK]b[UNK][UNK]
Actual : xr8rp



Prediction : [UNK][UNK][UNK]54
Actual : xwc43



Prediction : [UNK][UNK]43[UNK]
Actual : yp32g



사이즈가 커도 뚫을 수가 없다...

한 사이트에 최적화된 CAPTCHA를 만들어보면 어떨까?

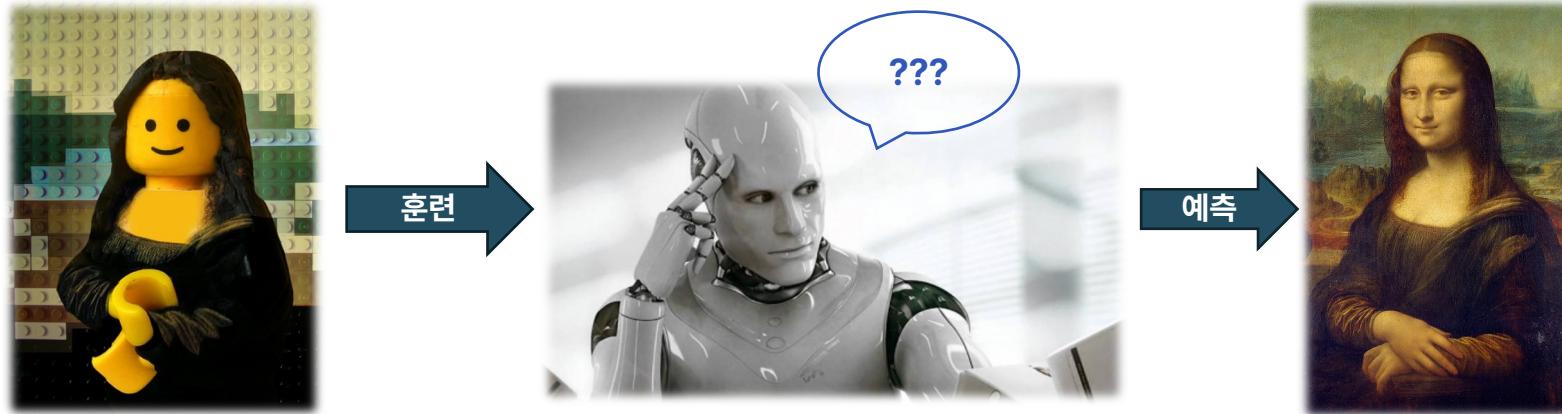
5

2차 시도: 특화 모델

한 CAPTCHA에 특화된 모델을 만들어보고 싶은데...

이미지를 일일이 다운받아 레이블을 달아서 훈련시키기는 번거롭다(귀찮다)...

이미지를 똑같이 따라서 만들면...
비슷한 효과를 낼 수 있지 않을까?



특화 Captcha 만들기

1차 시도 : DCInside Captcha 특화 모델

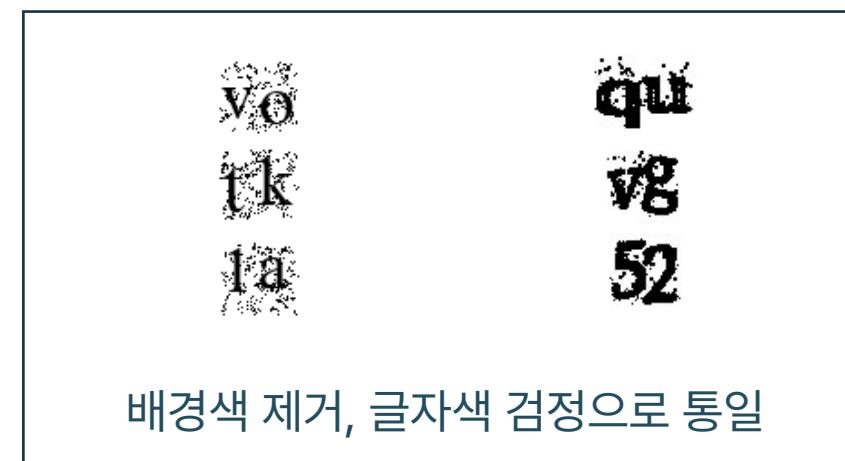


위 : 원본 이미지(예측 대상)
아래 : 모방 이미지(학습 데이터)

```

Epoch 12/100
141/141 [=====] - 15s 104ms/step - loss: 7.1589 - val_loss: 7.1883
Epoch 13/100
...
Epoch 99/100
141/141 [=====] - 15s 106ms/step - loss: 7.1424 - val_loss: 7.2043
Epoch 100/100
141/141 [=====] - 15s 108ms/step - loss: 7.1402 - val_loss: 7.2023
노이즈와 색상 등의 영향으로 학습 진행 X(7의 지옥에 갇힌 loss값...)

```



배경색 제거, 글자색 검정으로 통일

```

loss: 0.1625 - val_loss: 0.2505
loss: 0.1801 - val_loss: 0.1965

```

학습까지는 성공
과연 결과는?

특화 Captcha 만들기

한줄평 : 뭔가 되는 듯 했으나 실패!

학습 데이터와 실제 이미지 간의 간극이 大
→ 수많은 차이로 인해 예측 모델 정확도 ↓

→ 글자 수가 적다고 얇봤다가 큰 코 다쳤다...
조금 더 모방하기 쉬운 캡챠로 Retry!

예측 결과

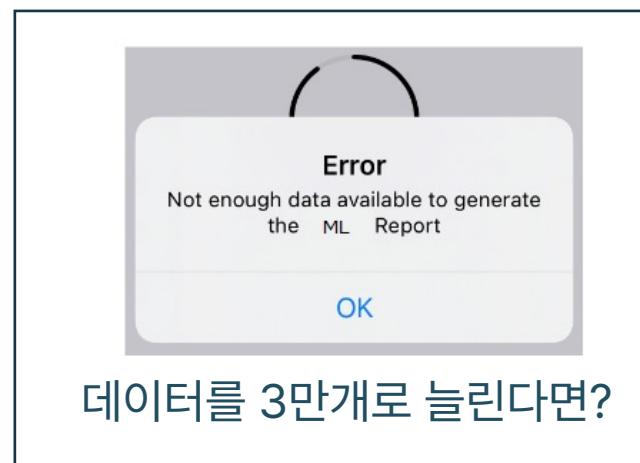
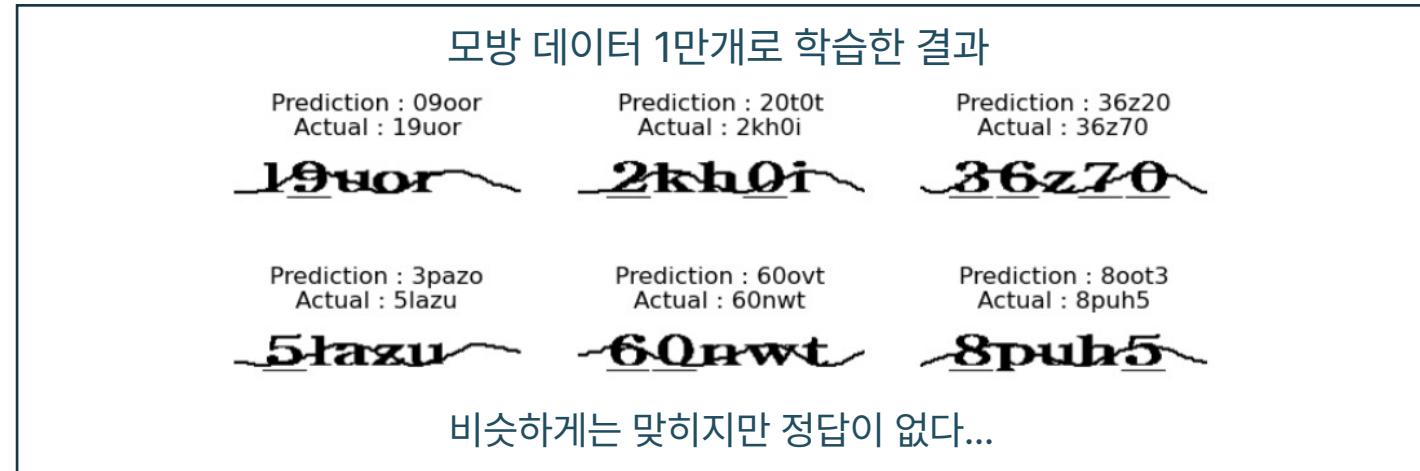
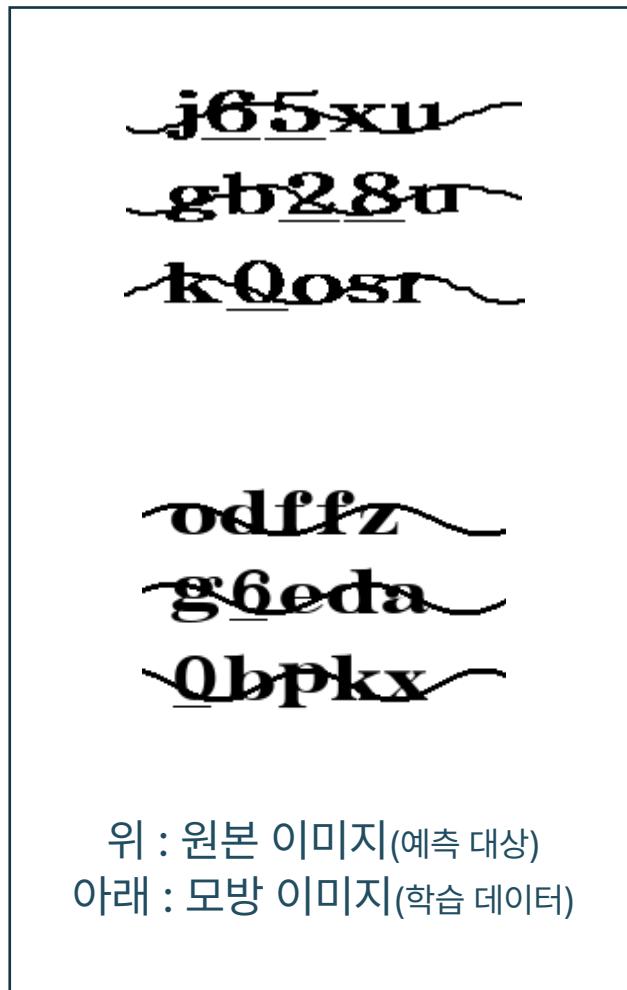
Prediction : 23 Actual : 23 	Prediction : 20 Actual : 3q 	Prediction : 48 Actual : 46 	Prediction : 90 Actual : 52 	Prediction : y4 Actual : 7d 
Prediction : qm Actual : 7n 	Prediction : 06 Actual : 8z 	Prediction : 9h Actual : 9b 	Prediction : 96 Actual : a6 	Prediction : 8n Actual : am 
Prediction : dn Actual : bh 	Prediction : ex Actual : cv 	Prediction : 81 Actual : e4 	Prediction : 9k Actual : mg 	Prediction : 69 Actual : mq 
Prediction : g4 Actual : n7 	Prediction : mb Actual : nh 	Prediction : 1r Actual : p5 	Prediction : gq Actual : pv 	Prediction : um Actual : qu 

보라 : 정답

주황 : 한 글자는 정답

특화 Captcha 만들기

2차 시도 : NETIS 공유기 CAPTCHA 특화 모델



과대적합을 방지하는 장치

Part 5 특화 Captcha 만들기

2차 시도 : NETIS 공유기 Captcha 특화 모델 결과 확인 및 피드백

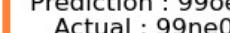
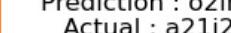
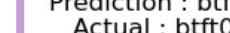
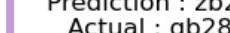
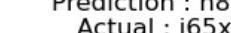
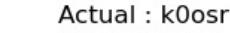
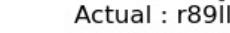
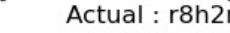
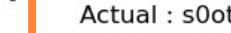
한줄평 : 아쉽긴 하지만 성과는 있다!

데이터를 추가하기만 했을 때 :
2개 정답 **but** 인식률 개선 효과 미미

가우시안 블러까지 추가했을 때 :
3개 정답 + 인식률 대폭 개선

→ 조금 더 잘 모방한다면 더 좋은 성과를 얻을 수 있으리라고 확신

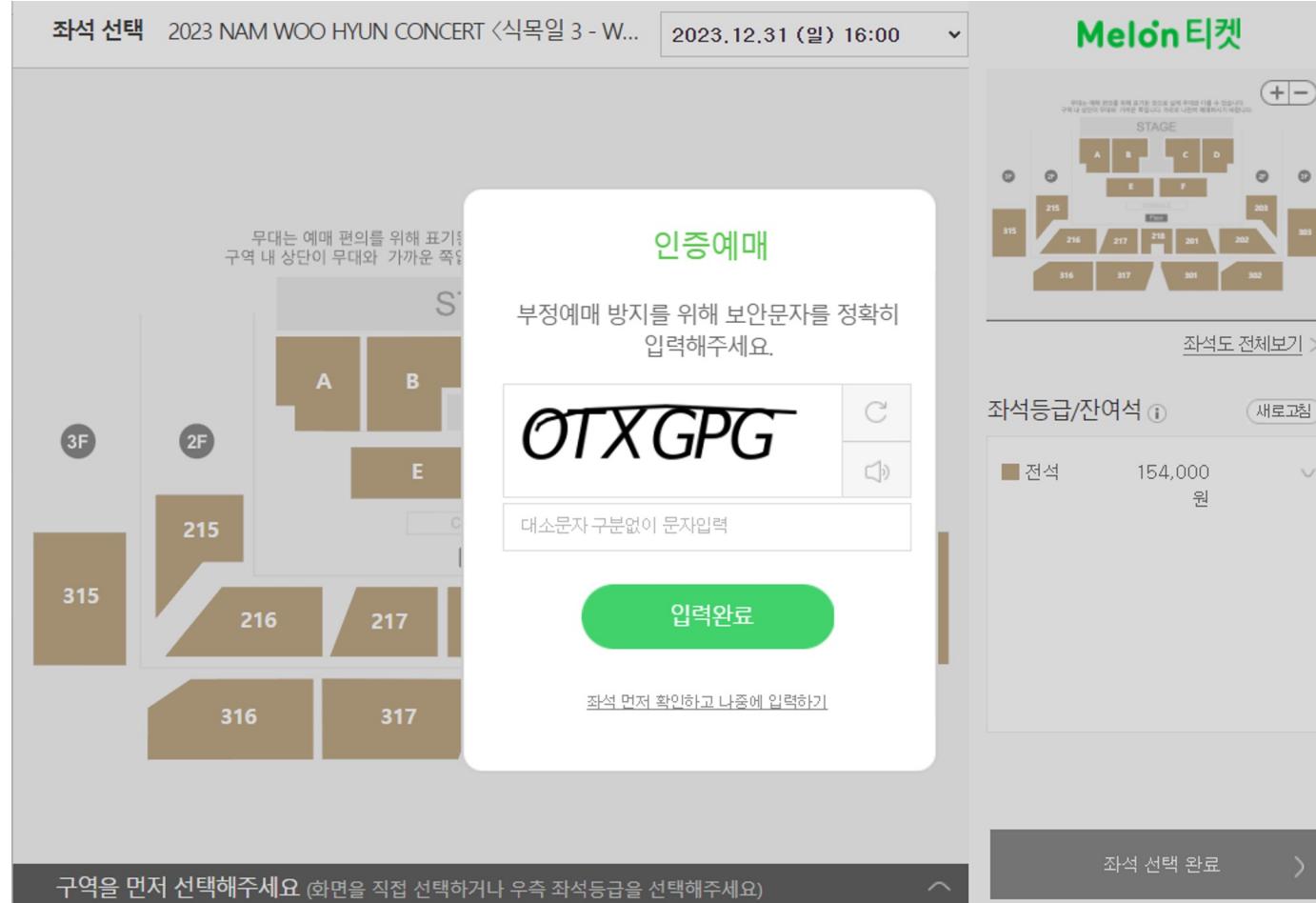
예측 결과

Prediction : 09bzl Actual : 09lgl 	Prediction : 13a2a Actual : 13a2a 	Prediction : l9eor Actual : 19uor 	Prediction : 2kb0i Actual : 2kh0i 	Prediction : 36z70 Actual : 36z70 
Prediction : 30tbv Actual : 30thj 	Prediction : 4qiar Actual : 4qiqqr 	Prediction : 54ozc Actual : 5lazu 	Prediction : 60oxt Actual : 60nwt 	Prediction : 8ocb5 Actual : 8puh5 
Prediction : 99oe0 Actual : 99ne0 	Prediction : o2ln2 Actual : a21j2 	Prediction : btft0 Actual : btft0 	Prediction : zb28c Actual : gb28u 	Prediction : n85xc Actual : j65xu 
Prediction : k0o5z Actual : k0osr 	Prediction : o94xb Actual : n94vb 	Prediction : v89l[UNK] Actual : r89ll 	Prediction : v8bo[UNK] Actual : r8h2m 	Prediction : s0otr Actual : s0otr 

보라 : 정답
주황 : 정답에 근접

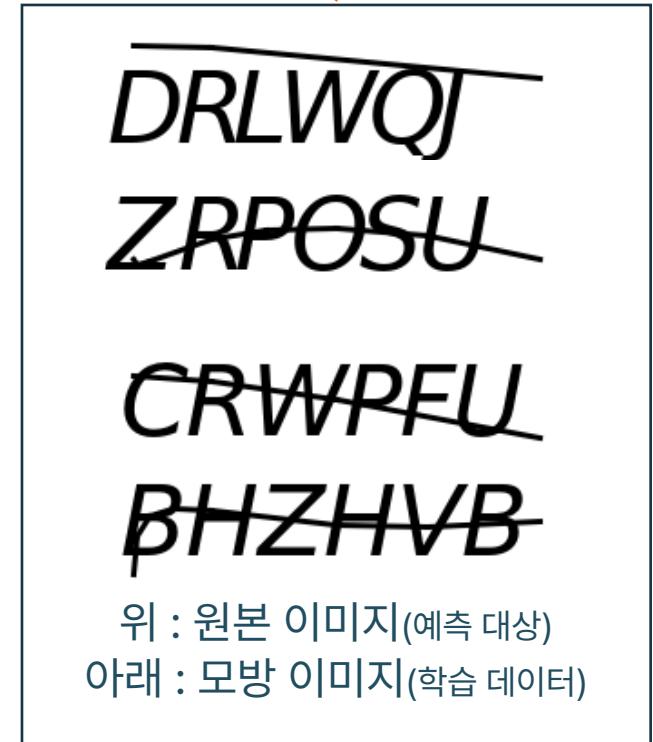
특화 Captcha 만들기

3차 시도 : Melon Ticket 특화 모델 및 매크로 제작



Melon Ticket에서 좌석 예약시 뜨는 실제 화면

구분이 되시나요?



1만 개 생성 + 가우시안 블러

Part 5 특화 Captcha 만들기

3차 시도 : Melon Ticket 특화 모델 결과 확인 및 피드백

한줄평 : 대성공! 놀라울 정도로 잘 맞힌다...

가우시안 블러처리 하지 않았을 때:
몇 개 정답이 있긴했으나 실망스러운 결과

가우시안 블러까지 추가했을 때 :
52개 샘플 중 38개 정답 = **정확도 73%**

→ 정확도를 조금 더 개선하고 **매크로** 제작을 시도

GaussianBlur 적용 유무의 차이

Prediction : XOYPXU
Actual : AOYPAD

~~AOYPAD~~

Prediction : XWGXD[UNK]
Actual : AWGZAD

~~AWGZAD~~

Prediction : CHIHII
Actual : BCHHHB

~~BCHHHB~~

Prediction : UDYUCU
Actual : BDYDCU

~~BDYDCU~~

Prediction : BVIFIT
Actual : BVIETG

~~BVIETG~~

Prediction : CSPUI[UNK]
Actual : CZZPUT

~~CZZPUT~~

Prediction : IUCPIB
Actual : DCPTBT

~~DCPTBT~~

Prediction : DRLWQJ
Actual : DRLWQJ

~~DRLWQJ~~

Prediction : THHXC[UNK]
Actual : FZRHAC

~~FZRHAC~~

Prediction : GLVOTF
Actual : GEVOTF

~~GEVOTF~~

Prediction : IIMSLY
Actual : HMSEYK

~~HMSEYK~~

Prediction : INTIXI
Actual : HTIXTK

~~HTIXTK~~

P : AOYPAD
A : AOYPAD

~~AOYPAD~~

P : AWGZAD
A : AWGZAD

~~AWGZAD~~

P : BCHHHB
A : BCHHHB

~~BCHHHB~~

P : BDYDCU
A : BDYDCU

~~BDYDCU~~

P : BVIETG
A : BVIETG

~~BVIETG~~

P : CZZPUT
A : CZZPUT

~~CZZPUT~~

P : DCPTBT
A : DCPTBT

~~DCPTBT~~

P : DRLWQJ
A : DRLWQJ

~~DRLWQJ~~

P : LZRHAC
A : FZRHAC

~~FZRHAC~~

P : GEVOTF
A : GEVOTF

~~GEVOTF~~

P : HMSEYK
A : HMSEYK

~~HMSEYK~~

P : HTIXTK
A : HTIXTK

~~HTIXTK~~

특화 Captcha 만들기

Melon Ticket 모델 정확도 개선

1Epoch마다 정확도를 계산



가장 높은 정확도를 기록하는 가중치를 Save

Epoch	Accuracy
20	0.923
21	0.577 ?
22	0.942 !
23	0.903
24	0.942 !
25	0.885

가장 좋은 가중치를 적용한 결과

Accuracy : 0.9423076923076923

Incorrect : ['JRLOKF', 'PEXOAF', 'XYMHYI']

P : AOYPAD A : AOYPAD	P : AWGZAD A : AWGZAD	P : BCHHHB A : BCHHHB	P : BDYDCU A : BDYDCU	P : BMZOAP A : BMZOAP
AOYPAD BVIETG	AWGZAD CZZPUT	BCHHHB DCPTBT	BDYDCU DRLWQJ	BMZOAP DWYZQ
P : FZRHAC A : FZRHAC	P : GEVOTF A : GEVOTF	P : HMSEYK A : HMSEYK	P : HTIXTK A : HTIXTK	P : HUGMAH A : HUGMAH
FZRHAC HZYZMT	GEVOTF IMMWFW	HMSEYK JNKBMT	HTIXTK JRLOKE	HUGMAH KOCNHY
P : LKATBU A : LKATBU	P : MQIWQV A : MQIWQV	P : MYGSTZ A : MYGSTZ	P : MYIMOA A : MYIMOA	P : NOXKAZ A : NOXKAZ
LKATBU OGNMUT	MQIWQV OJONMI	MYGSTZ PEXOAF	MYIMOA PKFEHV	NOXKAZ PWPJWI
P : QJXZYU A : QJXZYU	P : QTOZWG A : QTOZWG	P : RDPRXP A : RDPRXP	P : RECKMW A : RECKMW	P : RQXBZY A : RQXBZY
QJXZYU RVFHJJ	QTOZWG TSYWZM	RDPRXP UAPLFX	RECKMW UGCWHP	RQXBZY VDBKIK
P : VGRUPI A : VGRUPI	P : VSZNLE A : VSZNLE	P : UAPLFX A : UAPLFX	P : UGCWHP A : UGCWHP	P : VDBKIK A : VDBKIK
VGRUPI XJZAOV	VSZNLE XPEIJL	UAPLFX XPEIJL	UGCWHP XCVWMT	VDBKIK XEBHQP
P : XJZAOV A : XJZAOV	P : XPEIJL A : XPEIJL	P : XPTHVN A : XPTHVN	P : XCVWMT A : XCVWMT	P : XEBHQP A : XEBHQP
XJZAOV ZIEQMD	XPEIJL ZRPOSU	XPTHVN ZRPOSU	XCVWMT XTMHYI	XEBHQP YRVMKN

6

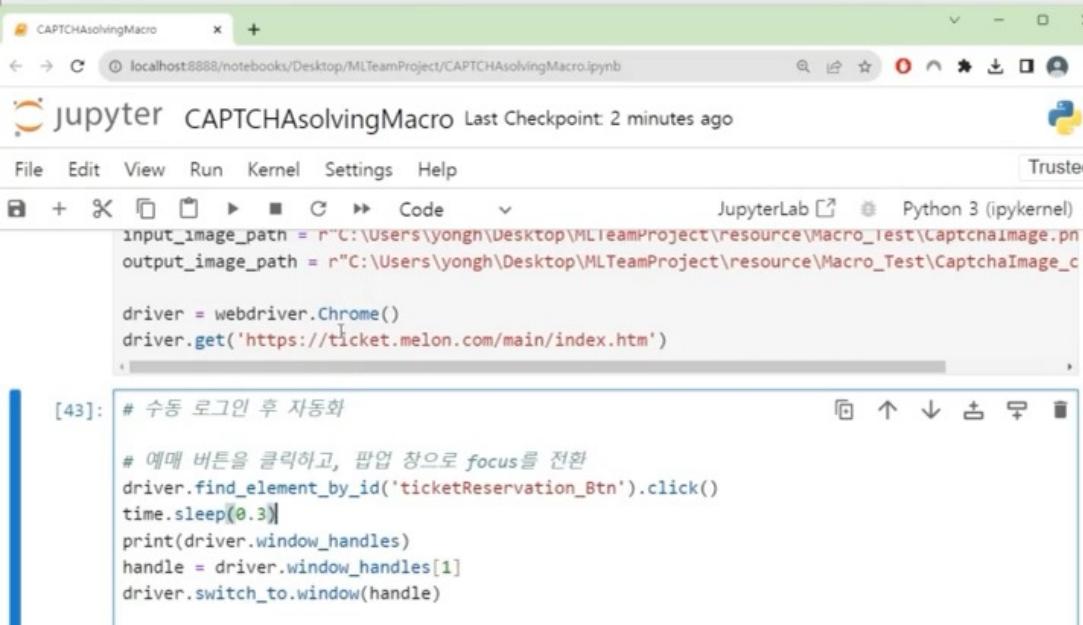
멜론티켓 매크로

멜론티켓 매크로(Powered by TensorFlow + Selenium Lib)

HUFS 기계학습 7조

<CRNN CAPTCHA Solving MACRO 시연>

수동 로그인 후 [예매하기] 버튼부터 자동!



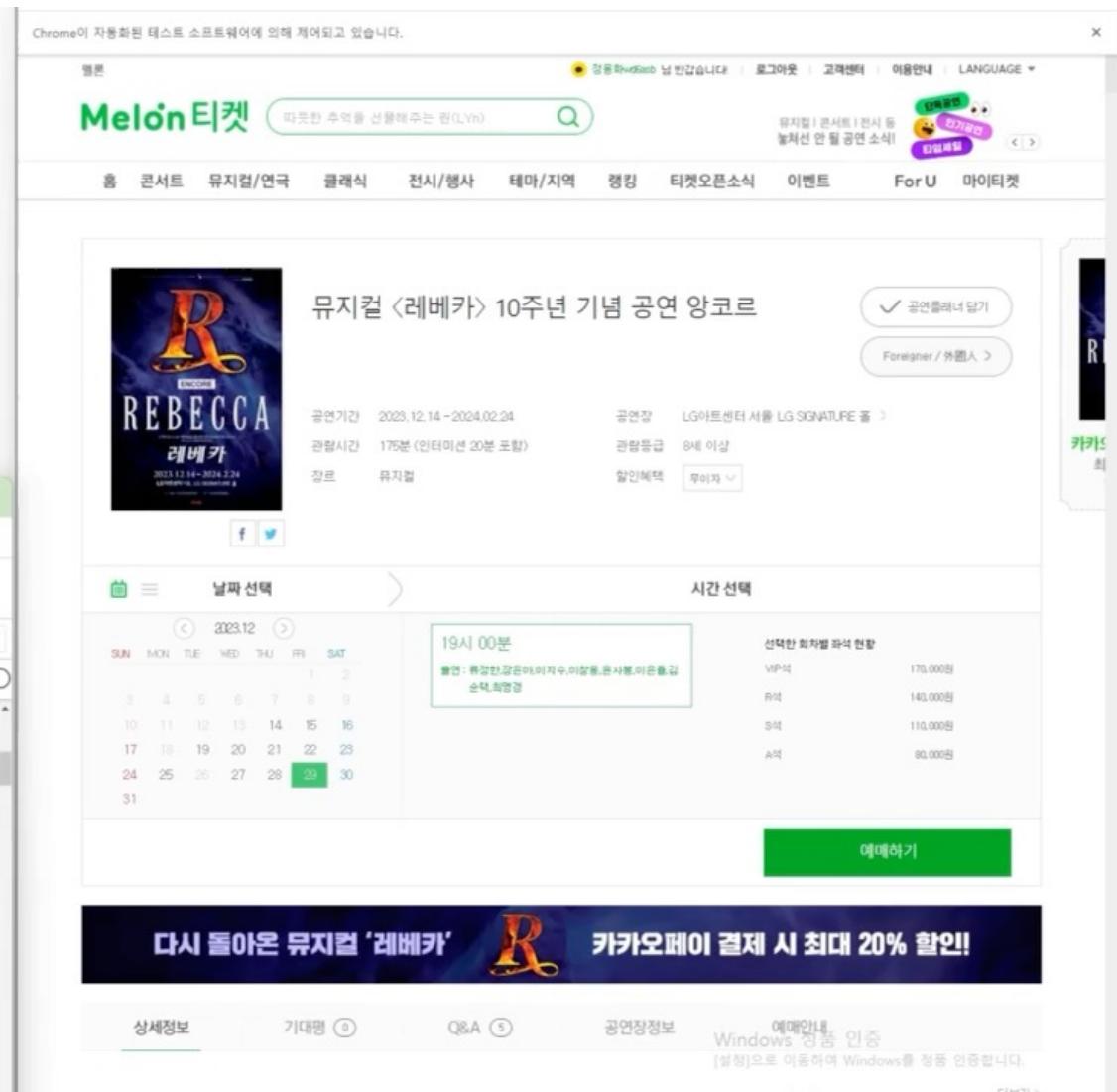
```

jupyter CAPTCHAsolvingMacro Last Checkpoint: 2 minutes ago
File Edit View Run Kernel Settings Help Trusted
+ □ ▶ ■ C Python 3 (ipykernel)


driver = webdriver.Chrome()
driver.get('https://ticket.melon.com/main/index.htm')

[43]: # 수동 로그인 후 자동화
# 예매 버튼을 클릭하고, 팝업 창으로 focus를 전환
driver.find_element_by_id('ticketReservation_Btn').click()
time.sleep(0.3)
print(driver.window_handles)
handle = driver.window_handles[1]
driver.switch_to.window(handle)

```



Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

Melon 티켓 따뜻한 추억을 선물해주는 티켓(Lyn)

뮤지컬 <레베카> 10주년 기념 공연 앙코르

공연기간: 2023.12.14 ~ 2024.02.24
관람시간: 175분 (인터미션 20분 포함)
장르: 뮤지컬

관람등급: 8세 이상
할인혜택: 무이자

날짜 선택: 2023.12.29 19:00분

선택한 회차별 좌석 현황						
VIP석	170,000원					
석	140,000원					
석	110,000원					
A석	90,000원					

예매하기

다시 돌아온 뮤지컬 '레베카' **R** 카카오페이지 결제 시 최대 20% 할인!

상세정보 기대평 Q&A 공연장정보 예매안내 Windows 인증

[설정]으로 이용하여 Windows를 정품 인증합니다.

시연을 위한 동영상입니다.

7

보안 측면에서의 고찰

내가 Captcha를 뚫는 사람이라면...

1. 범용적인 모델을 제작하는 것은 난해하다!

폰트, 노이즈의 종류 등 Captcha마다 그 변수가 다양하다.

결국, 특정 Captcha에 적합한 모델을 만들어보자



CRWPFL

이 노이즈도 SimpleCaptcha Lib으로 만들었어요!

2. 오픈소스 라이브러리를 사용하는지 확인해보자!

멜론티켓의 경우 SimpleCaptcha를 사용한다. 이처럼 오픈소스 Captcha를 사용하는 서비스인지 확인해 보는 것도 좋다.

3. 노이즈나 블러를 활용하여 과적합을 줄여보자!

모델이 과적합되지 않도록 Noise나 Blur을 주는 것은 정확도 향상에 많은 도움이 된다.

내가 Captcha 매크로를 방어하는 사람이라면...

1. 오픈소스 라이브러리에 의존하지 말자!

2. Captcha 데이터를 다양하게 구성하자!

최대한 많은 변수를 활용하는 것이 방어를 효과적으로 할 수 있다.

대소문자 중 하나만 쓰는 것보단 혼용 하는 것을, 기왕이면 숫자도 추가하자

3. 오답 비중이 높은 문자를 전략적으로 사용하자.

숫자 '0'과 'o', 'C'/'c', 'r'/'v'와 같이 오답이 발생하는 문자를 더 높은 빈도로 등장하게 하는 것이 도움이 될 수 있다.

8

참고문헌

1. 이미지 출처

<https://www.pinterest.co.kr/pin/642044490653600077/> 레고 모나리자

<https://ko.wikipedia.org/wiki/%EB%AA%A8%EB%82%98%EB%A6%AC%EC%9E%90> / 모나리자

<https://gall.dcinside.com/mgallery/board/write/?id=aoegame> 디씨 캡챠

<https://support.withings.com/hc/en-us/community/posts/4957303667089-Health-report-not-enough-data-> / not enough to... 이미지

<https://ticket.melon.com/main/index.htm> / 멜론티켓

https://keras.io/examples/vision/captcha_ocr/ keras crnn

혼자서 공부하는 딥러닝(한빛출판사, 박해선) / CNN, RNN 설명

2. 논문

'An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition' (Baoguang Shi)

3. 사용한 주요 라이브러리

Pillow, TensorFlow(Keras), OpenCv, Selenium, Simple-Captcha(Java)